

Express mail Label #:EL290559065US

SPECIFICATION

IBM Docket No. STL9-2000-0063

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that We, Laurence E. England of Morgan Hill, California and citizen of the United States, Doctor Howard J. Glaser of San Jose, California and citizen of the United States, and Doctor Robert D. Moyer of Morgan Hill, California and citizen of the United States, have invented new and useful improvements in

METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING AN E-BUSINESS CONNECTOR BUILDER

of which the following is a specification:

1
2
3 **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR**
4 **PROVIDING AN E-BUSINESS CONNECTOR BUILDER**
5
6
7
8
9

10 A portion of the Disclosure of this patent document contains material which is subject
11 to copyright protection. The copyright owner has no objection to the facsimile reproduction by
12 anyone of the patent document or the patent disclosure, as it appears in the Patent and
13 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to transaction processing systems, and more particularly to adapting a transaction-based application to process transactions over a network such as an internet or intranet.

2. Description of the Related Art

In an era of e-business, brick and mortar companies transitioning to the e-business world must retrofit/adapt existing mainframe applications to support customer to business transactions (C2B) as well as business to business transactions (B2B) on the web. These companies have their key applications and data on mainframes as a result of both heritage and the reliability/performance afforded by these systems. To utilize their current mainframe applications and data stores, IT shops need programming techniques to pass transactions from web application servers to back end applications residing on the mainframe. Many of these applications use transaction processing subsystems such as the IBM® Customer Information Control System (CICS®) or the IBM® Information Management System (IMS) in conjunction with programming languages such as COBOL or PLI. (IBM® and CICS® are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.) In these systems, the transaction processor runtimes initiate/start programs written in languages, such as COBOL, via pragmatically callable transaction identifiers (ID). The transaction identifiers have associated with them both the program to be called and a definition of the data to be passed to and from the program. The definition and declaration of the data to be passed to and from programs is called the communication area or COMM Area. It is typically contained in COBOL programs and COBOL COPYBOOKS. Through this method of linking together programs and passing shared data, elaborate applications have been built. Unfortunately, these transaction applications have evolved over many years and have

1 been frequently modified. The applications have become large and unwieldy. Many of the
2 original creators have long since retired and have taken knowledge about how these
3 applications work and where to find key parts. Finding these transactions and COMM Area
4 definitions is required to link web technology to mainframe applications.

5
6 To further compound the problem, different programming styles and techniques were
7 used in developing these programs. Some programs were written with the transactions neatly
8 separated from the presentation logic (this is called External Call Interface; used in
9 Client/Server CICS product literature or ECI), while others used techniques combining both
10 the 3270 presentation display code with the transaction logic (this called External Presentation
11 Interface; used in Client/Server CICS product literature or EPI). The former ECI method is
12 easily adaptable to connectors and adapters while the latter EPI method is more troublesome
13 since transactions are bound up in the display logic.

14
15 In summary, the problems faced by conventional methods in bringing mainframe
16 applications out to the web are:

17 finding usable ECI transactions where the business logic has been already separated
18 from the 3270 presentation logic;

19 extracting the key parameter definitions for communicating information to and from
20 these transactions;

21 identifying which of these parameters are input, output, input/output, and unreferenced;

22 transferring this information to a workstation from a mainframe to build connectors;

23 and

24 pairing connection information with the parameter information.

25
26 Conventional methods have failed to provide solutions to these problems. Thus, there
27 is a clearly felt need for a method of, system for, article of manufacture for, and computer
28 program product for providing an improved adaptation of a transaction-based application to
29 process transactions over a network such as an internet or intranet..

SUMMARY OF THE INVENTION

The present invention comprises a method, system, and article of manufacture for adapting a transaction-based application to process transactions over a network such as an internet or intranet. The transaction-based application is adapted by scanning the source code of the transaction-based application to identify a transaction and information related to the transaction; storing the related information in a database; extracting from the database parameter definitions describing communication of information by the transaction; identifying a parameter usage type for each parameter, said parameter usage type selectable from the parameter usage type set comprising input, output, input/output, and unreferenced; displaying the transaction and a subset of the related and extracted information; allowing a user to select the transaction; and using the identified and extracted information to package the user-selected transaction in a form compatible with a connector building tool.

One aspect of a preferred embodiment of the present invention provides that the compatible form comprises a parsable file containing information which can be parsed by a connector building tool.

Another aspect of a preferred embodiment of the present invention generates a documentation file describing the parsable file.

Another aspect of a preferred embodiment of the present invention provides that the documentation file comprises field description information and connection information.

Another aspect of a preferred embodiment of the present invention uses the identified and extracted information to build a connector.

Another aspect of a preferred embodiment of the present invention uses the identified

1 and extracted information to build an enterprise Java bean connector. (Java and all Java-based
2 trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or
3 both.)
4

5 Another aspect of a preferred embodiment of the present invention queries the database
6 to find program parts comprising the transaction-based application and to identify relationships
7 between the program parts.
8

9 Another aspect of a preferred embodiment of the present invention provides that the
10 related information is a member of the set comprising relationships, call hierarchies,
11 transactions, communication areas, parameters, the flow of data elements, and resources
12 employed.

13 A preferred embodiment of the present invention has the advantage of providing
14 improved adaptation of a transaction-based application to process transactions over a network
15 such as an internet or intranet.
16

17 A preferred embodiment of the present invention has the further advantage of providing
18 an improved user interface for adapting a transaction-based application.
19

20 A preferred embodiment of the present invention has the further advantage of providing
21 improved usability in a tool for adapting a transaction-based application.
22

23 A preferred embodiment of the present invention has the further advantage of
24 providing improved integration in a tool for adapting a transaction-based application.
25

26 A preferred embodiment of the present invention has the further advantage of providing
27 improved functionality in a tool for adapting a transaction-based application.
28

1 A preferred embodiment of the present invention has the further advantage of finding
2 transactions.

3
4 A preferred embodiment of the present invention has the further advantage of extracting
5 key parameter definitions for communicating information by a transaction.

6
7 A preferred embodiment of the present invention has the further advantage of
8 identifying which of these key parameters are input, output, input/output, and unreferenced.

9
10 A preferred embodiment of the present invention has the further advantage of
11 transferring this information to a workstation from a mainframe to build connectors.

12
13 A preferred embodiment of the present invention has the further advantage of
14 associating transaction connection information with transaction parameter information.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the Description of the Preferred Embodiment in conjunction with the attached Drawings, in which:

Figure 1 is a block diagram of a distributed computer system used in performing the method of the present invention, forming part of the apparatus of the present invention, and which may use the article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to practice the present invention;

Figure 2 is a block diagram of a preferred embodiment of the present invention;

Figure 3 is a flowchart illustrating the operations preferred in carrying out the preferred embodiment of the present invention; and

Figure 4 through Figure 9 illustrate graphical user interfaces preferred in carrying out the preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. A person skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will be apparent to a person skilled in the relevant art that this invention can also be employed in a variety of other devices and applications.

Referring first to **Figure 1**, there is depicted a graphical representation of a data processing system **8**, which may be utilized to implement the present invention. As may be seen, data processing system **8** may include a plurality of networks, such as Local Area Networks (LAN) **10** and **32**, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Alternatively, networks **10** and **32** may be intranets or portions of the internet. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network. Each said network may also consist of a plurality of processors coupled via a communications medium, such as shared memory, shared storage, or an interconnection network. As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16** and may be provided with a pointing device such as a mouse **17**.

The data processing system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be preferably coupled to LAN **10** by means of communications link **22**. The mainframe computer **18** may also be coupled to a storage device **20** which may serve as remote storage for LAN **10**. Similarly, LAN **10** may be coupled via communications link **24** through a sub-system control unit/communications controller **26** and

1 communications link 34 to a gateway server 28. The gateway server 28 may be an IWS which
2 serves to link LAN 32 to LAN 10. Preferably, server 28 is a web application server which
3 passes transactions from a requester 30 on the internet 32 to the mainframe 18 upon which a
4 back-end application serving the transaction is executing.

5
6 With respect to LAN 32 and LAN 10, a plurality of documents or resource objects may
7 be stored within storage device 20 and controlled by mainframe computer 18, as resource
8 manager or library service for the resource objects thus stored. Of course, those skilled in the
9 art will appreciate that mainframe computer 18 may be located a great geographic distance
10 from LAN 10 and similarly, LAN 10 may be located a substantial distance from LAN 32. For
11 example, LAN 32 may be located in California while LAN 10 may be located within North
12 Carolina and mainframe computer 18 may be located in New York.

13
14 Software program code which employs the present invention is typically stored in the
15 memory of a storage device 14 of a stand alone workstation or LAN server from which a
16 developer may access the code for distribution purposes, the software program code may be
17 embodied on any of a variety of known media for use with a data processing system such as a
18 diskette or CD-ROM or may be distributed to users from a memory of one computer system
19 over a network of some type to other computer systems for use by users of such other systems.
20 Such techniques and methods for embodying software code on media and/or distributing
21 software code are well-known and will not be further discussed herein.

22
23 As will be appreciated upon reference to the foregoing, it is often desirable for a user
24 to link a transaction-based application on the mainframe 18 to the internet 32 and/or World
25 Wide Web (web), where the transaction-based application was not originally designed for web
26 or internet based transactions. It is also desirable for this user to perform this application
27 development on a workstation 12 in lieu of performing the application development on the
28 mainframe 18 itself. A preferred embodiment of the present invention provides capabilities to
29 assist such a user in performing such an adaptation of the transaction-based mainframe

1 application to enable it to process transactions over a network such as the internet or web.

2
3 Referring now to **Figure 2** illustrating a block diagram of a preferred embodiment of
4 the present invention, the preferred embodiment comprises two key components:

- 5 • an application knowledge database **200**, preferably stored in an IBM® DB2 ®
6 relational database; and
- 7 • a tool comprising a user interface **205** which displays transaction information and an
8 analysis/packaging tool **210** which packages user selected transactions in a form
9 consumable for web component connector building tools **215**. (IBM® and DB2® are
10 registered trademarks of International Business Machines Corporation in the United
11 States, other countries, or both.)

12
13 The application knowledge base **200** is loaded by a source scanner **220** which scans
14 (process block **310**) the application source code **225** to find relationships **230**, call hierarchies
15 **235**, transactions **240**, communication areas **245**, parameters **250**, the flow of data elements
16 **255**, and resources employed **295**, such as database table access, transient data queues, and
17 basic mapping support (BMS) maps. Once loaded (process block **315**) into preset database
18 schemas, the database **200** can be queried to find program parts and identify relationships
19 between program parts.

20
21 The invention contained herein is an application which uses the information stored in
22 the database **200** (process block **320**) to display callable ECI transactions contained within
23 specified CICS Regions (process block **330**), provides an indication as to the usage of
24 parameters, either input, output, input/output, or unreferenced (process block **325**), and creates
25 a file **260** in a form which can be consumed by a connector building tool **215** such as the IBM
26 Enterprise Access Beans (EAB) or the Microsoft COMTI Builder (process block **340**).
27 (Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or
28 both.)

1 Referring now to Figure 5, the left hand tabbed pane **505** displays CICS regions **510** in
2 a tree **515** hierarchically decomposing the transactions (**520** and **525**) into Load Module **530**,
3 Source code **535**, and Copy Books **540**. The right hand tabbed page **545** offers various views
4 on the parts of the transaction components such as the COMM Area **550** shown in the right
5 hand pane **545** of **Figure 5** (process block **330**). This not only shows the parameters **555** that
6 are usable from this transaction **520**, but also the parameter type **560** as to whether they are
7 input, output, input/output, or unreferenced parameters (process block **325**). The information
8 display (process block **330**) is based on the information stored (process block **315**) in the
9 knowledge base database **200**. In the case of the nomination of input, output, input/output, or
10 unreferenced by the database **200** and the tool **210**, the tool **210** lets the user override those
11 assignments **560**. When an ambiguity exists, a pessimistic view of input/output is taken.

12
13 Other views include the original source code **565**, the CICS Linkage section **570** of the
14 COBOL program, CICS Queues **575**, and a Call Graph **580** depicting transaction and program
15 call relationships. This information is constructed via under the cover queries by the tool **210**
16 against the knowledge relation database **200**.

17
18 Referring now to **Figure 4**, the tool **210** is loaded by a dialog **410** which, via a query
19 **265** to the database **200**, displays machine/link pack area (LPAR) names **420** and CICS regions
20 **430** contained within. This dialog **410** is found under the File item of a task bar and is labeled
21 Import Region. Once a machine/LPAR **420** and Region **430** are selected, a query **265** is issued
22 against the database **200** to retrieve a list of CICS ECI transactions and their parts contained in
23 the selected region (process block **320**). The returned information **270** is used to construct the
24 tree view **515** and populate the editors and views on the right hand pane **545** as illustrated in
25 **Figure 5** (process block **330**).

26
27 After the tool **210** is loaded with the information **270** on a CICS region , the user can
28 select transaction IDs to be used in a connector **275** (process block **335**) and the tool will
29 construct (process block **340**) two files per transaction: a communication area file **260** and a

documentation file **280**. The communication area file **260** contains an ASCII-readable COMM Area file which can be parsed by a connector builder application to build a connector (process block **345**), such as those which import COBOL and COPYBOOK files. The documentation file **280** provides documentation about the communication area file **260** including field description information **285** and connection information **290** such as OS/390 machine, LPAR, CICS Region name. Both the communication area file and the documentation file **280** may be implemented as a meta-language file, preferably as an Extensible Markup Language (XML) file. Both of these files will automatically be given a name by combining the region name and the transaction ID. Selecting and building these files is via a wizard comprising the sequence of display screens **600**, **700**, **800**, and **900** as depicted in the window sequences illustrated in **Figures 6, 7, 8, and 9**.

The wizard first displays the view **600** comprising a list **610** of the user-selected transactions for which the communication area file and documentation file will be generated. Initially, this list **610** is empty as the user has not selected any transactions. Clicking on the Forward button **620** to proceed with the process causes the wizard to display the view **700** which allows the user to add or remove transactions from the selected list. The left hand view **710** lists the transactions **720** in the CICS region, and the right hand view **730** lists those transactions **740** selected by the user. After the user has selected the transactions, view **800** is displayed which now comprises a list **810** of the user-selected transactions for which the communication area file and documentation file will be generated. Clicking on the Forward button **820** to proceed with the process causes the wizard to display the view **900** which allows the user to select a storage location **910** for the generated files.

Referring now to **Figure 3**, the flowchart illustrates the operations preferred in carrying out the preferred embodiment of the present invention. In the flowcharts, the graphical conventions of a diamond for a test or decision and a rectangle for a process or function are used. These conventions are well understood by those skilled in the art, and the flowcharts are sufficient to enable one of ordinary skill to write code in any suitable computer programming

1 language. The process 300 of adapting a transaction-based application to process transactions
2 over a network such as an internet or intranet begins at process block 305. Thereafter, process
3 block 310 scans the source code of the transaction-based application to identify the transaction
4 and the related information. Process block 315 then stores the related information in a
5 database. From this database, process block 320 extracts parameter definitions describing
6 communication of information by the transaction. For each parameter, process block 325
7 identifies a parameter usage type, said parameter usage type selectable from the parameter
8 usage type set comprising input, output, input/output, and unreferenced. The transaction and a
9 subset of the related and extracted information is then displayed by process block 330. From
10 such display, process block 335 allows a user to select the transaction. For such a user-selected
11 transaction, process block 340 uses the identified and extracted information to package the
12 user-selected transaction in a form compatible with a connector building tool, and in process
13 block 345 the connector building tool uses the identified and extracted information to build a
14 connector. Thereafter, the process ends at process block 350.

15
16 Using the foregoing specification, the invention may be implemented using standard
17 programming and/or engineering techniques using computer programming software, firmware,
18 hardware or any combination or sub-combination thereof. Any such resulting program(s),
19 having computer readable program code means, may be embodied within one or more
20 computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,
21 semiconductor memories such as Read-Only Memory (ROM), Programmable Read-Only
22 Memory (PROM), etc., or any memory or transmitting device, thereby making a computer
23 program product, i.e., an article of manufacture, according to the invention. The article of
24 manufacture containing the computer programming code may be made and/or used by
25 executing the code directly or indirectly from one medium, by copying the code from one
26 medium to another medium, or by transmitting the code over a network. An apparatus for
27 making, using, or selling the invention may be one or more processing systems including, but
28 not limited to, central processing unit (CPU), memory, storage devices, communication links,
29 communication devices, servers, input/output (I/O) devices, or any sub-components or

1 individual parts of one or more processing systems, including software, firmware, hardware or
2 any combination or sub-combination thereof, which embody the invention as set forth in the
3 claims.

4
5 User input may be received from the keyboard, mouse, pen, voice, touch screen, or any
6 other means by which a human can input data to a computer, including through other programs
7 such as application programs.

8
9 One skilled in the art of computer science will easily be able to combine the software
10 created as described with appropriate general purpose or special purpose computer hardware to
11 create a computer system and/or computer sub-components embodying the invention and to
12 create a computer system and/or computer sub-components for carrying out the method of the
13 invention. Although the present invention has been particularly shown and described with
14 reference to a preferred embodiment, it should be apparent that modifications and adaptations
15 to that embodiment may occur to one skilled in the art without departing from the spirit or
16 scope of the present invention as set forth in the following claims.